

1. (CURRENTLY AMENDED): A graphics processing method, comprising the steps of:

(a.) caching texture memory fetches, using a cache tag assignment which is [[essentially]] unique mapped, the cache tag assignment having a tag length and mip mapping addresses, while

(b.) generating condensed cache tags, by removing two bits from the tag length by means of a remapping which exploits [[the ]]different address resolutions implied by level of detail settings in [[the ]]different mip mapping processes to re-encode the mip mapping addresses

(c.) and using [[said ]]the condensed tags for [[said ]]caching step (a.);

(d.) coding a map level identifier so that

a largest map level uses 1 bit to designate a map level and  $((m-i) + (n-j))$  bits to specify said addresses on said x- and y-axis,

a second largest map level uses 3 bits to designate a map level and  $((m-i) + (n-j))$ -2) bits to specify the addresses on the x-axis and y-axis, and

successively smaller map levels use greater than 3 bits to designate a map level and less than  $((m-i) + (n-j))$ -2) bits to specify the addresses on the x-axis and y-axis.

2. (CANCELED)

3. (CURRENTLY AMENDED): A graphics processing method, comprising caching

texture memory fetches using a cache tag assignment in which a unique relation between a mip-mapping-level-of-detail parameter and coordinate bits defines a smaller tag address for any given memory address;

wherein a map level identifier is coded so that

a largest map level uses 1 bit to designate a map level and  $((m-i) + (n-j))$  bits to specify said addresses on said x- and y-axis,

a second largest map level uses 3 bits to designate a map level and  $((m-i) + (n-j))$ -2) bits to specify the addresses on the x-axis and y-axis, and

successively smaller map levels use greater than 3 bits to designate a map level and less than  $((m-i) + (n-i)-2)$  bits to specify the addresses on the x-axis and y-axis.

4. (CURRENTLY AMENDED): The graphics processing method of Claim 3, wherein [[said ]]the cache tag assignment is generated by combining a mip-map-level-of-detail parameter which can have at least  $2^{J-1} + 1$  different values together with coordinate bits which can have as many as  $2^K$  different values into fewer than  $J + 2K$  bits without loss of information; wherein J represents the number of bits for [[the ]]a level of detail and K represents the number of bits for arbitrary coordinate values.

5. (CURRENTLY AMENDED): The graphics processing method of Claim 3, wherein [[said ]]the cache tag assignment is generated by combining a first parameter which can have at least  $2^{J-1} + 1$  different values together with coordinate bits which can have as many as  $2^K$  different values into fewer than  $J + 2K$  bits without loss of information;

wherein [[said ]]the first parameter and [[said ]]the coordinate bits are three-dimensional coordinates; and

wherein J represents the number of bits for [[the ]]a level of detail and K represents the number of bits for arbitrary coordinate values.

6. (CURRENTLY AMENDED): A method of generating condensed cache tags, comprising the steps of:

- (a.) concatenating[[ the]] a texel address on[[ the]] an x- and y-axis with a map level identifier, where addresses on the x-axis can require m bits, addresses on the y-axis can require n bits, and [[said ]]the map-level identifier can require p bits;
- (b.) if two caches are being used for odd/even maps, deleting [[the ]]a least significant bit of [[said ]]the map level identifier;
- (c.) if texels are being stored in the cache in  $2^i \times 2^j$  patches, such that i and j represent a texel's address on the x- and y-axis respectively and said address having i and j least significant bits, deleting the i least significant bits of the address on the x-axis and the j least significant bits of the address on the y-axis; and
- (d.) coding said map level identifier so that
  - [[the ]]a largest map level uses 1 bit to designate [[the ]]a map level and  $((m-i) + (n-j))$  bits to specify said addresses on said x- and y-axis,
  - [[the ]]a second largest map level uses 3 bits to designate [[the ]]a map level and  $((m-i) + (n-j)-2)$  bits to specify [[said ]]the addresses on [[said ]]the x-axis and y-axis, and
  - successively smaller map levels use greater than 3 bits to designate [[the ]]a map level and less than  $((m-i) + (n-j)-2)$  bits to specify [[said ]]the addresses on [[said ]]the x-axis and y-axis.

7. (CANCELLED): ~~A cache system for a texture map, comprising:~~

~~a texture memory containing at least one map, wherein [[the ]]addresses for said map can require m bits for [[the ]]an x-axis coordinate, n bits for [[the ]]a y-axis coordinate, and p bits for [[the ]]a map level identifier; and~~  
~~a direct mapped texture cache for [[said ]]the texture memory, configured to be accessed using lookup tags which require m + n - 1 or fewer bits.~~

8. (CURRENTLY AMENDED): A graphics processing method, comprising the steps of:

- (a.) caching texture memory fetches, using a cache tag assignment which is  
[[essentially]] unique mapped, while
- (b.) generating condensed cache tags, by means of a remapping which exploits  
[the ]]different address resolutions implied by level of detail settings in  
[[the ]]different mip mapping processes to re-encode [[the ]]mip mapping  
addresses into a length which is only one bit longer than [[the ]]a  
maximum condensed length of [[the ]]a spatial [[addresses]] address
- (c.) and using [[said ]]the condensed tags for [[said ]]caching step (a.)
- (d.) coding a map level identifier so that  
a largest map level uses 1 bit to designate a map level and  $((m-i) + (n-j))$  bits to  
specify said addresses on said x- and y-axis,  
a second largest map level uses 3 bits to designate a map level and  $((m-i) + (n-j))$ -  
2) bits to specify the addresses on the x-axis and y-axis, and  
successively smaller map levels use greater than 3 bits to designate a map level  
and less than  $((m-i) + (n-j))$ -2) bits to specify the addresses on the x-axis  
and y-axis.